International Journal of Scientific Engineering and Technology
Volume No.5 Issue No.9, pp: 430-432

ISSN:2277-1581
01 September 2016

Secure Authentication for Online Banking Against Man-In-The-Browser Attack Using Least Significant Bit Embeddings.

UMUHOZA Leoncie ,Dr.Cheruiyot W.K, Dr.Ann Kibe
School of Computer Science and Information Technology,Department of computing,
Jomo Kenyatta University of Agriculture and Technology
Email:umuhozaleonciee@gmail.com

Abstract: ***MitB attacks are a sophisticated new hacking technique associated with Internet crime, especially that which targets customers of Internet banking and operate by stealing authentication data and altering legitimate user transactions to benefit the attackers. This paper presents LSB embeddings approach for enhancing user authentication during an online transaction to tackle the problem of MitB attack.***
Keywords: Man-in-the-browser, Authentication, Online Banking, LSB Embedding.

I. Introduction

Online banking is an electronic payment system that enables customers of a financial institution to conduct financial transactions on a website operated by the institution, such as a retail bank, virtual bank, credit union or building society. Online banking is also referred as internet banking, e-banking, virtual banking and by other terms (Gandy, 1995). The MitB attack leverages what is known as a Trojan horse (or simply a Trojan). A Trojan is malicious software that is somehow installed often initiated by various social engineering tactics and resides concealed on the user's computer, frequently undetectable by traditional virus scanning (Entrust, 2014). MitB attack Malware today which has become the method of choice to attack financial institutions and it is very dangerous malware because of its ability to hide from anti-virus software, installs itself as part of the victim's client itself (i.e,the browser) ,it can modify details of the transaction that you initiate, It can read your identity, bank balance, banking passwords, debit/Credit, card numbers and session keys. It modifies a user's content when an online banking site is visited by adding extra fields to the page in order to compromise second authentication mechanisms (Prince et al., 2010). MitB malware is mostly undetectable by current antivirus software, although it may be detected if protection levels are set very high, which would also inhibit many innocuous programs. For all intents and purposes, the injected page looks like an original page served by the bank and can truly challenge even the most sophisticated, security-aware end users (Alcorn et al., 2014).Normally internet browser allows external browser extension to be installed in order to add additional features to the browser. This extension allows user to have extended functionalities other than common browser function. However, by allowing external extension to be installed with or without the user's consent into the end user machine through the browser it makes it more vulnerable to MitB attacks.

In fact, attackers normally use the same distribution channel as legitimate extension to distribute the malicious plug-in into the user's browser. Moreover, with lack of knowledge in security awareness, end user would not be able to differentiate the genuine or malicious extension. Consequently, malicious plug-in are able to infiltrate into the browser and would be able to manipulate any sensitive information between a client and a browser. Once inside these Trojans spy on the browser sessions and become active as soon as they detect some activity on the financial sites mentioned in their configuration file. Once Activated these Trojan can intercept all the data sent and received between the user and the bank server, all this happens in the background and the user is not aware that his security has been compromised. Indeed, as the method of cracking the security code get more complex and powerful. There is need to develop more powerful security application. These powerful applications allow the user to work on entrusted computers confidently.

II. Material and Methodology

A. Proposed LSB Algorithm

The LSB algorithm is a well known algorithm and it has been extended to make data hiding less detectable and more secure. They can embed lossless, preserving all information about the data, or the data may be generalized so that it takes up less space. The LSB is one of the main techniques in spatial domain (Akathar et al., 2013). The technique converts image into shaded Gray Scale image, this image will act as reference image to hide the text and using this grey scale reference image any text can be hidden (Kavitha, 2012). Single character of a text can be represented by 8-bit. If the reference image and the data file are transmitted through network separately, the research can achieve the effect of Steganography. Any huge amount of text material can be hidden using a very small image. To decipher the text is not possible neither intercepting the image or data file separately, so it is more secure (Bassam et al., 2012).
Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values.

ISSN:2277-1581
01 September 2016

1) Embedding Algorithms

In this technique, to randomized the cover image, random key is used and then hides the secrete bit of the message into the cover image using least significant bit method. Stego key and random key is shared by transmitting and receiving ends. The random-key is usually used to seed a pseudo random number generator to select pixel locations in an image for embedding the secret message.

Input: Cover Image, Secrete message and stego key

Ouput: Stego image

1) Read character from text file that is to be hidden and convert the ASCII value of the character into equivalent binary value into an 8 bit integer array.
2) Read the RGB colour image (cover image) into which the message is to be embedded.
3) Read the last bit of red pixel.
4) Initialize the random key and randomly permute the pixels of cover image and reshape into a matrix.
5) Initialize the stego-key and XOR with text file to be hidden and give message.
6) Insert the bits of the secret message to the LSB of the Red plane's pixels.
7) Write the above pixel to Stego Image File.

Embeddings algorithms are shown in Figure 1.0:

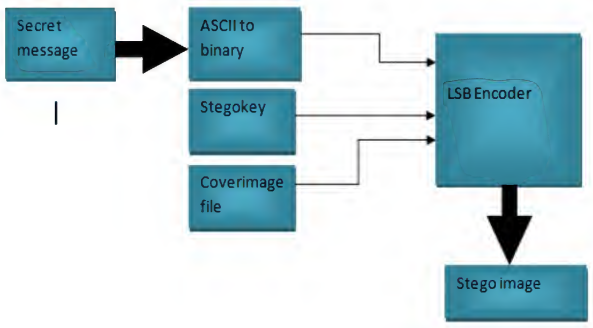

Figure 1.1 Embeddings algorithm

2) Extraction Algorithm

Input: Stego Image file, Stego-key and randomkey

Output: Secrete Message

1) Open the Stego image file in read mode and from the Image file, read the RGB colour of each pixel.
2) Extract the red component of the host image.
3) Read the last bit of each pixel.
4) initialize the random-key that gives the position of the message bits in the red pixel that are embedded randomly.
5) For decoding, select the pixels and Extract the LSB value of red pixels.
6) Read each of pixels then content of the array Converts into decimal value that is actstegaidlly ASCII value of hidden character.
7) ASCII values got from above are XOR with stego-key and gives message file, which we hide inside the cover image.

Extraction algorithms are shown in Figure 1.1:

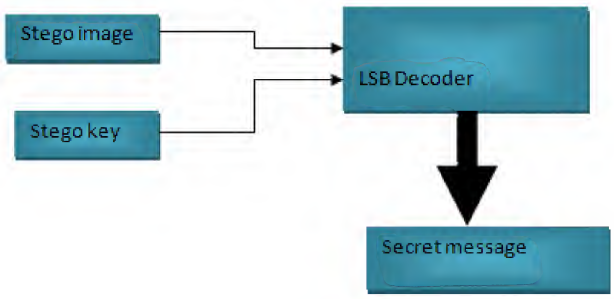

Figure 1.2 Extraction algorithms

The approach will help to hide secret data of users in a cover medium in such a way that the MitB attack cannot perceive it,And it will be used to hide user secrecy data, their information into image of .bmp format and send the image from a source to a destination on online banking

III. RESULTS AND TABLES

A. *Concealing Message*

The proposed approach is designed for BMP images. It first compares the length of the message to be concealed with the size of the image to ensure that the image can hold the secret file. If the size of secret file is more, then a new image is selected. When using a 24 bit color image, a bit of each of the red, green and blue colour components can be used, so a total of 3 bits can be stored in each pixel. So one layer between R, G, B is selected and message is inserted in the selected layer. Thus, an 800 × 600 pixel image can contain a total amount of 800x600x1=480.000 bits (60.000 bytes) of secret data. It has three levels of security as follows.

Level 1: The message is inserted at a random pixel value of the image as inserted by the sender. It can be any row and column of the image matrix. But precaution must be taken such that message length should not exceed matrix size.

Level 2: The message to be sent is encrypted using an encryption algorithm.

Level 3: The encrypted message is now inserted to image using LSB technique. In LSB technique encrypted message is converted to binary form and inserted in the least significant bit of pixel value as inserted before. These the three level of security enable the process to be a highly secure message system. If anyone tries to break into the system then he has to know the starting position of the message then encryption method used and method of insertion. Till he/she got all information the value of information might have been lost.

B. *Experimental results*

1) Embedding result

In this section, the proposed algorithm has been implemented. As shown in Figure 2.1 Steganography at Sender side shows the process of encoding a bit to hide information in randomly selected cover image.

ISSN:2277-1581
01 September 2016

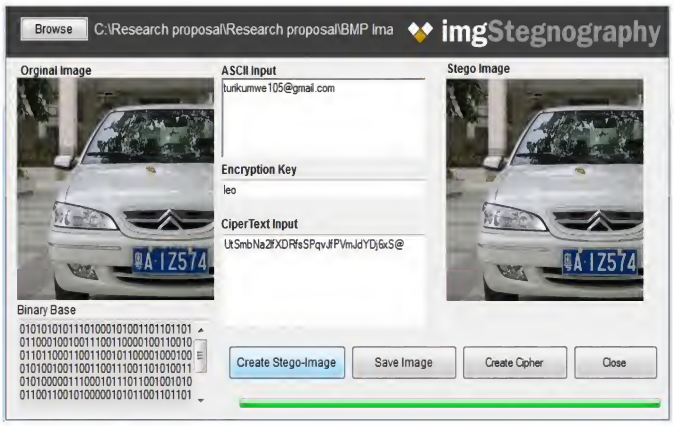

Figure 2.1 Steganography at Sender side

2) Receiver Result

Figure 2.2 Output at Receiver side shows the process of decoding an image to uncover the hidden information under it. It can be seen that after authentication takes place and the user selects the decode image option.

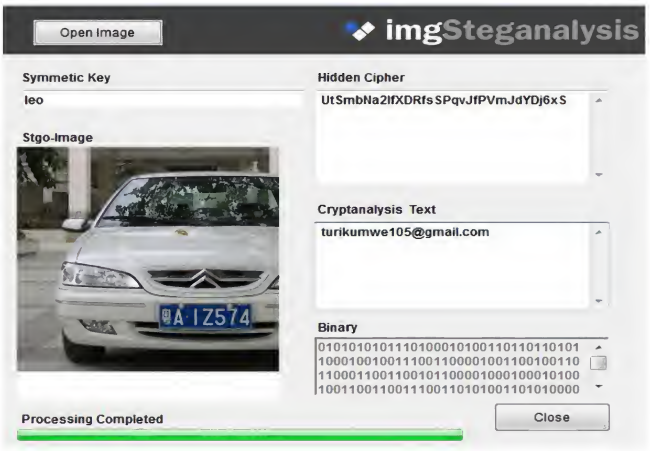

Figure 2.2 Output at Receiver side

C. Discussion

As shown in figure 2.1 Stego image look alike the original image which does not show any distortion. Thus the stego image will not attract attention towards itself. So it can be transferred to the recipient without displaying information within itself.

D. Performance analysis

1) PSNR (Peak Signal to noise Rotio)

PSNR as a metric computes the peak signal-to-noise ratio, in decibels, between two images. It is used in steganography to Measure the peak signal-to-noise ratio in the original image and the stego image after embedding the hidden data. The quality difference between the cover image and the stego-image is measured through PSNR. As shown in Figure 2.3 PSNR Analysis, larger the PSNR, higher is the image quality

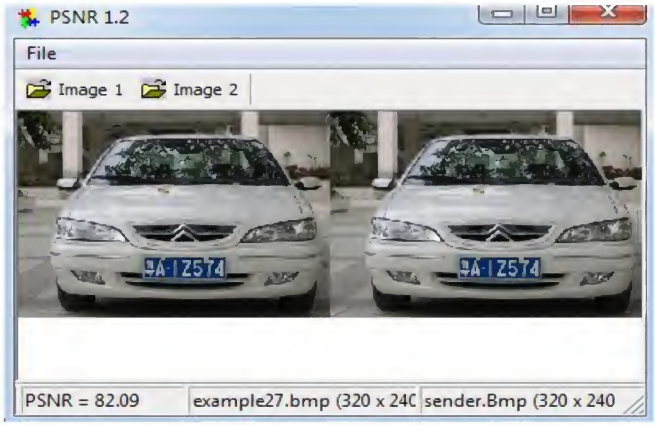

Figure 2.3 PSNR Analysis between original and stego image

As shown in Figure 2.3 PSNR ratio results is 82.09 db which mean s images are better of quality to secure message.

IV. Conclusion

Security in the online banking is like arms compete between the cybercriminals and the information securers; with proposed approach user unique authentication password in connection to the bank is hidden inside a cover image using the LSB steganography where user authentication information is placed above the cover image in its original form.

Acknowledgement

I would like to express my special appreciation to my supervisors, Dr.Cheruiyot W.K and Dr.Ann Kibe for their valuable guidance and advice that enabled me to successfully complete this study.

References

i. *Akhtar, P.j. (2013).Enhancing the security and quality of LSB based image steganography.*

ii. *Alcorn, F. (2014).The Browser Hacker's Handbook*

iii. *BarYosef,N.(2010).TheEvolutionofProxyTrojans.retrievedfrom http://www.securityweek.com/evolution-proxy-trojans*

iv. *Bassam Jamil Mohd,S.A.-H.(2012).FPGA Hardware of the LSB steganography Method.*

v. *Beaver, K., & Shaw,J.(2011).; Multifactor Authentication for Dummies,USA*

vi. *Entrust.(2014).DefeatingMan-in-the-Browser.Retrievedfrom https://www.entrust.com/.../2014/.../WP_Entrust-MITB_March2014.pdf*

vii. *Gandy, T. (1995).Banking in e-space, the banker, 145 (838), pp. 74–76.*

viii. *Prince, B. (2010).Understanding Man-in-the-Browser Attacks Targeting Online Banks*

ix. *Richard E. Smith. (2001). Authentication: From Passwords to Public Keys. Addison Wesley.*

x. *RSA. (2012).Making Sense of Man-in-the-browser Attacks: Threat Analysis and Mitigation for Financial Institutions, retrieved from http://www.emc.com/security/rsa-securid.html.*